Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences

Hochschule für Technik und Wirtschaft

Fachbereich Wirtschaftswissenschaften II

Studiengang Angewandte Informatik

Masterarbeit

Konzeption und Prototyp einer intelligenten Arbeitssuche nach persönlichen Fähigkeiten

---

vorgelegt von Sergej Meister
Matrikelnummer: s0521159

Erstgutachter: Prof. Dr. Christian Herta

Zweitgutachter: Prof. Dr.-Ing. habil. Dierk Langbein

Berlin, 24. Januar 2016

Vorwort

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Inhaltsverzeichnis

1. Einleitung

1.1 Motivation

Was versteht man unter dem Begriff *Information*? Was sind *Daten*? Die beiden Fragen sind sehr beliebt in der Informatikstudium, besonders im ersten Semester. Doch die Informationen und Daten spielen nicht nur in der Informatik sondern in allen Gebieten eine wichtige Rolle. Täglich treffen die Menschen eine Entscheidung auf der Basis Ihres Wissensdaten, die aus den Informationsmengen analysiert und interpretiert werden. Dadurch dass die Daten aus den Informationen interpretiert werden, hängt die Entscheidungsfaktor wesentlich von dem *„Qualität*" den bereitgestellten Informationen ab.

In der heutigen Informationsgesellschaft durch das schnelle Wachstum von digital verfügbaren Informationen ist es für die Menschen schwieriger geworden, die entscheidungsrelevante Daten aus den Informationsmengen zu filtern [Beringer 2012]. Deswegen bietet das Internet viele verschiedene Suchportals für unterschiedliche Interessengruppe, die eine gezielte Suche nach bestimmten Informationsmengen erlauben. Einer davon sind Job Portals.

Das Job Portal unterstützt ihrer Anwender rund um das Thema *„Arbeitsuche*" und *„Bewerbungsprozess*". Ein Arbeitssuchender legt Suchkriterien, wie Berufsbeizeichnung und Region, fest und bekommt in weniger Sekunde eine Liste von Arbeitsangeboten, die den Suchkriterien entsprechen. Es ist offensichtlich, dass die Arbeitssuche dank Job Portals viel effizienter und einfacher geworden. Trotzdem muss der Arbeitssuchender immer noch viele Informationen selbst raus finden und interpretieren. Zum Beispiel für die Entscheidung, ob die Stelle den gewünschten Anforderungen entspricht, sind die Daten wichtig, wie Berufsname, Firmenname und Qualifikation. Für das Bewerbungsschreiben werden noch weitere Daten benötigt, wie Kontaktperson, Mailadresse, Firmenhomepage und Firmenadresse. Diese Informationen muss der Arbeitssuchender ständig selbst in einem Arbeitsangebot finden und interpretieren. Genau mit dieser Problematik beschäftigt sich das Programm *„Intellijob*", das die Daten automatisch aus dem Arbeitsangebot extrahiert.

Das Programm *„Intellijob*" wurde im Rahmen der Veranstaltung *„Informationssystemen*" unter Betreuung von Professor Christian Herta entwickelt und kann bereits Berufsbezeichnung, Mailadresse, Firmenhomepage, Kontaktperson und Firmenadresse automatisch auslesen und in Datenbank speichern. Außerdem verfügt das Programm über eine Weboberfläche, die die gespeicherte Daten tabellarisch

darstellt.

Der weitere Nachteil von Job-Portals ist die Implementierung von der Suche, die die in der Profile gespeicherte personalisierte Daten nicht berücksichtigt. Denn die typische Suchkriterien eines Job Portals sind einfache Stichworte wie Softwareentwickler, Java, C♯ u.s.w. Daraus wird es ersichtlich, dass die Suche nach einem konkreten Beruf erfolgt. Dieses Suchverfahren eignet sich gut für Quereinsteiger und für die Menschen, die auf der Suche nach neuer Berufsorientierung sind, aber nicht für die Menschen, die eine Arbeit entsprechend ihren Fähigkeiten und Qualifikationen suchen. Wäre es nicht besser, wenn die Suche nach persönlichen Fähigkeiten und Qualifikationen erfolgte, die von Benutzer selbst bewertet werden? Nehmen wir an, der Mensch bewertet seine Java-Kenntnisse mit 2 und C♯ mit 5 Sternen. Dann ist es doch logisch, dass alle C♯ Arbeitsangebote in der Ergebnisliste zuerst angezeigt werden müssen. Vielleicht ist es auf dem ersten Blick nicht ganz ersichtlich, woran die beide Verfahren sich unterscheiden. Die Suche nach einem Beruf als Softwareentwickler kann aber die Java- Arbeitsangebote zuerst anzeigen, was aus der Sicht der Arbeitssuchender mit den besseren C♯ Kenntnissen nicht ganz zutreffend ist. Der weitere Unterschied liegt im Entscheidungsfaktor. Bei dem ersten Verfahren entscheidet der Mensch, welche Arbeit er haben will. Bei dem zweiten Verfahren entscheidet das Programm, welche Arbeit für den Menschen am besten geeignet ist.

Die Untersuchung von mehreren Job-Portals wie *„Xing"*, *„StepStone"*, *„JobScout24"*, *„Monster"* hat bestätigt, dass es tatsächlich kein deutsches Job-Portal gibt, das die Suche nach persönlichen Fähigkeiten und Qualifikationen unterstützt.

1.2 Ziele der Arbeit

Im Rahmen dieser Masterarbeit wird ein Suchverfahren implementiert, dass die Suche nach persönlichen Fähigkeiten und Qualifikationen unterstützt. Die Suche muss mit *Elasticsearch*[1] realisiert und in der Anwendung *IntelliJob* integriert werden. Um die Suchqualität bewerten zu können, müssen mehrere unterschiedliche Suchqueries gebildet und genau untersucht werden. Am Ende der Arbeit wird geprüft, ob das neue Suchverfahren gegenüber der traditionelle Suche nach Stichworte eine bessere Informationsmenge liefert. Das Suchverfahren wird als effektiv bewertet, wenn die erste Arbeitsangebote in der Ergebnisliste den persönlichen Fähigkeiten und Qualifikationen besser entsprechen werden.

Das weitere Thema dieser Arbeit wird die Datenbeschaffung sein. Um nach persönliche Fähigkeiten und Qualifikationen suchen zu können, müssen diese Daten zuerst erfasst werden. Dafür wird das Projekt *„Intellijob"* um eine neue Web-Form erweitert, die die Bewertung von persönlichen Fähigkeiten und Qualifikationen ermöglicht. Die Daten, die dem Arbeitssuchender bereitgestellt werden, müssen sinnvoll gruppiert werden, so dass die Kriterien wie Java und C♯ unter den Sammelbegriffen Programmieren, Softwareentwickeln, gefunden werden könnten. Der

[1] **www.elastic.co**.

Arbeitssuchender muss in der Lage sein, seine eigene Daten speichern, editieren und löschen zu können. Außerdem darf der Benutzer selbst entscheiden, welches Suchverfahren eingesetzt werden soll.

1.3 Aufbau der Arbeit

Der schriftliche Teil der Arbeit beschreibt das komplette Entwicklungsprozess. Im nächsten Kapitel *„Grundlagen"* werden die Technologie erläutert, die für die Entwicklung neues Suchverfahrens relevant sind. Dabei wird das Projekt *IntelliJob* und die eingesetzte Technologie kurz vorgestellt. Das Kapitel *„Systementwurf"* befasst sich mit Softwarearchitektur und beschreibt die Logikverteilung und die Datenkommunikation innerhalb der Anwendung. Das vierte Kapitel *„Entwicklung und Implementierung"* stellt das Prototyp des neuen Suchverfahrens vor. Dabei wird sowohl das Datenmodel als auch die wichtigste Funktionen detailliert beschrieben. Außerdem wird in diesem Kapitel auch die API-Schnittstelle dokumentiert. Im Unterkapitel *„Benutzeroberfläche"* wird die Web-Form zur Erfassung von persönlichen Fähigkeiten und Qualifikationen als Bild dargestellt. Das Kapitel *„Evaluierung"* vergleicht und bewertet das neue Suchverfahren. Problemen und Schwierigkeiten, die während der Implementierung aufgetreten sind, sowie die Art und Weise, wie diese Probleme gelöst oder eventuell nicht gelöst wurden, werden in dem 6. Kapitel erfasst. Zum Schluss wird die Arbeit zusammengefasst und ein Resümee gezogen. Ein Ausblick zeigt auf, welche Erweiterungen für die Anwendung in Zukunft noch sinnvoll sein könnten.

2. Grundlagen

In diesem Kapitel wird auf die allgemeinen und informatischen Grundlagen eingegangen, welche zum Verständnis dieser Masterarbeit relevant sind. Die Anwendung *IntelliJob* persistiert bereits die Daten in eine dokumentenorientierte Datenbank *MongoDB*[1] und es wird noch eine weitere dokumentenorientierte Technologie *Elasticsearch* eingesetzt. Deswegen ist es wichtig mindestens die Grundlagen sowie die Vor- und Nachteile von dokumentenorientierten Datenbanken im Blick zu behalten. Die in der Arbeit verwendete *Elasticsearch-Feature's* werden in einem extra Kapitel ausführlich beschrieben. Da die Suchfunktionalität in der bereits vorhandene Anwendung integriert werden muss, ist es für die Bewertung der Arbeit notwendig den IST-Zustand zu dokumentieren.

2.1 Dokumentenorientierte Datenbanken

Bei dokumentenorientierten Datenbanken werden die Daten in Form von Dokumenten gespeichert. Die typische Dokument-Formate sind XML, YAML und JSON. Die einzelnen Datenfelder werden als Key-Value Stores zusammengefasst. Jedes Dokument ist vollkommen frei bezüglich seine Struktur und das ist ein wesentlicher Unterschied zu relationalen Datenbanken, wo die Datenstruktur vordefiniert ist [Dorschel 2015]. Die Abbildung 1. zeigt eine typische normalisierte Datenstruktur in der relationalen Datenbanken. Einige Datensätze in der Tabelle *skills* beinhalten kein *"description"*. Die Spalte muss in dem Fall einfach leer bleiben. Es besteht keine Möglichkeit nur für bestimmte Datensätze auf die Spalte *"description"* zu verzichten.

category_id	name
1	Languages
2	Knowledges
3	Personal strengths

skill_id	category_id	name	description
1	1	German	
2	1	English	
3	2	Java	programming language java
4	2	PHP	programming language php

Abbildung 2.1: Relationale Tabellen *skill-categories* und *skills*

In der dokumentenorientierten Datenbanken ist es aber durchaus möglich, dass einige Dokumente innerhalb einer *Collection* kein *description* Feld haben(siehe Abbildung 3).

[1] **www.mongodb.org**.

```
{
  "_id" : ObjectId("569ab61a44ae6344028b6cb5"),
  "name" : "Languages"
},
{
  "_id" : ObjectId("569ab65644ae6344028b6cb7"),
  "name" : "Knowledges"
},
{
  "_id" : ObjectId("569ab66d44ae6344028b6cb9"),
  "name" : "Personal strengths"
}
```

Abbildung 2.2: Mongo Collection *skill-categories*

```
{
  "_id" : ObjectId("569ab85144ae6344028b6cbf"),
  "category" : {
     "_id" : ObjectId("569ab61a44ae6344028b6cb5"),
     "name" : "Languages"
  },
  "skills" : [
     {
        "_id" : ObjectId("569ab8dd44ae6344028b6cc0"),
        "name" : "deutsch"
     },
     {
        "_id" : ObjectId("569ab8dd44ae6344028b6cc1"),
        "name" : "english"
     }
  ]
},
{
  "_id" : ObjectId("569ab9d644ae6344028b6cc9"),
  "category" : {
     "_id" : ObjectId("569ab65644ae6344028b6cb7"),
     "name" : "Knowledges"
  },
  "skills" : [
     {
        "_id" : ObjectId("569ab9d644ae6344028b6cca"),
        "name" : "java",
        "description" : "programming language java"
     }
  ]
}
```

Abbildung 2.3: Mongo Collection *skills*

Wenn man die Abbildungen 2 und 3 genau anschaut, fehlt es sofort auf, dass die *Category-Daten* nicht redundant sind. In der relationalen Datenbanken werden die Datenattribute durch die Anwendung von verschiedenen Normalisierungsregeln solange zerlegt, bis keine vermeidbaren Redundanzen mehr vorhanden sind. Die dokumentenorientierte Datenbanken verfolgen den Prinzip der *Aggregation*. Der Begriff kommt eigentlich aus dem Domain-Driven Design und bedeutet, dass jedes Dokument eine eigenständige Einheit ist, das von keinen anderen beziehungsweise von möglichst wenigen Dokumenten abhängig ist. Dadurch können die Daten immer vollständig automar gespeichert oder gelesen werden.

Diese hohe Datenflexibilität ist aber nicht umsonst. Zum Ersten ist es oft aufwendig die Daten zu ändern, die bereits in anderen Dokumenten gespeichert sind. Ein Beispiel wäre dafür das Dokument mit dem Name *„Languages"* in der *Collection „skill-categories"* (Abbildung 2) mit einem neuen Attribute zu erweitern oder komplett zu löschen. In dem Fall muss diese *„Category"* auch in allen anderen Dokumenten gefunden und entsprechend geändert werden(Abbildung 3). Zum Zweiten können dokumentenorientierte Datenbanken aufgrund ihrer Schemafreiheit keine einfache Datenvalidierung der Attributen und Datentypen durchführen. Das muss stattdessen im Programmcode behandelt werden, dass zu höheren Softwarekosten führt und fehleranfällig ist [Jan 2013].

Wenn man das ganze kurz wörtlich zusammenfasst: Schemafreiheit, keine Datenvalidierung, mehr Programmcode, Fehleranfällig und aufwändige Datenänderung auf der Datenbankebene, fragt man sich, wieso werden aber die dokumentenorientierte Datenbanken, die seit 2008 unter dem Titel *„NoSQL"*[2] bekannt sind, immer beliebter? Eine einfache Antwort lautet, dass die NoSQL-Datenbanken oft schneller als relationale Datenbanken sind. Statt mehrere Tabellen mit JOIN-Operation abfragen zu müssen, führt ein NoSQL-Datenbank nur eine einzige Abfrage aus, um ein kompletter Datensatz zu erhalten. Die Datenkonsistenz beim Lesen und Schreiben in mehrere Tabellen wird in klassischen relationalen Datenbanken durch Transaktion-Mechanismen realisiert. Jede Transaktion ist isoliert von einander und wird ganz oder gar nicht ausgeführt. Wenn eine Transaktion in eine Tabelle schreibt, kann keine andere Transaktion auf die Tabelle zugreifen, bis der Schreibvorgang komplett abgeschlossen oder unterbrochen wird. Das führt dazu, dass die Lese- und Schreiboperation von parallel ausgeführten Transaktionen einander blockieren und warten müssen, bis alle nötige Ressource freigegeben werden. Dadurch dass NoSQL Datenbanken keine Abfrage blockieren, werden die auch schneller abgearbeitet. Es besteht zwar die Möglichkeit, dass inkonsistente Daten gelesen werden, aber aufgrund, dass die Daten oft nur aus einer einzigen *Collection* gelesen werden, ist die Fehlerwahrscheinlichkeit enorm klein im Vergleich zu viel steigender Performance[Dorschel 2015]. Die einzige Schutzmaßname, die von meisten NoSQL-Datenbanken unterstützt werden, ist die Datenversionierung. Bei jedem Speicherzugriff wird die Version des Dokumentes geprüft, ist die Version identisch dem bereits gespeicherten Dokument, dann wird ein *Exception* geworfen. Denn es ist ein Zeichen dafür, dass Dokument bereits von einem anderen *Request*

[2] **https://de.wikipedia.org/wiki/NoSQL**.

geändert wurde. Außerdem wird auch die Datenversionierung für die Synchronisierung zwischen mehrere Rechnerknoten innerhalb eines *Clusters* verwendet. Das *Sharding Concepts* in dokumentenorientierten Datenbanken wird im Kapitel über *Elasticsearch* detailliert beschrieben.

Fazit: Die dokumentenorientierte Datenbanken eignen sich besonders gut für die Anwendungen, wo hohe Geschwindigkeit bei Datenzugriffe und Datenverarbeitung im Vordergrund steht und die Datenredundanz wird nicht hoch priorisiert. Die meisten Datenbankzugriffe in der Anwendung *IntelliJob* sind Lesezugriffe. Es gibt wenige Möglichkeiten die Daten durch Benutzeraktion zu ändern. Die Änderungen beeinflussen nur eine einzige *Collection* und oft sogar nur ein einziges Dokument. Deswegen ist die Entscheidung über den Einsatz von dokumentenorientierten Datenbanken in der Anwendung richtig.

2.2 IntelliJob

Die Anwendung *„IntelliJob"* wurde bereits mehrmals erwähnt und muss endlich vorgestellt werden. Es gab mal eine Idee die bewerbungsrelevante Daten aus Arbeitsangeboten automatisiert zu extrahieren. Diese Idee wurde mit dem Programm *„IntelliJob"* realisiert. Da die Anwendung im Rahmen der Veranstaltung *„Informationssystemen"* eigenständig entwickelt und dokumentiert wurde, werden einige Inhalte aus der Arbeit in diesem Kapitel zitiert.

Bevor die Daten aus dem Arbeitsangebot gelesen werden könnten, muss dieses Arbeitsangebot erstmals ermittelt werden. Dafür werden mehrere Job-Agenten auf Job-Portals *Stepstone*[3] und *Monster*[4] angelegt, die die Arbeitsangebote per Email täglich senden. Das Email beinhaltet mehrere HTML-Links mit unterschiedlichen Arbeitsangeboten. Die Anwendung *„IntelliJob"* holt diese Emails aus dem Postfach ab, ruft die Links auf und speichert das geladene Inhalt in Datenbank ab.

Im nächsten Schritt werden die bewerbungsrelevante Daten, *Berufsbezeichnung*, *Firmenadresse*, *Firmenhomepage*, *Mailadresse* und *Kontaktperson*, aus dem Arbeitsangebot extrahiert.

Berufsbezeichnung wird einfach aus dem HTML-Link ausgelesen, das als *Value* immer eine Berufsbezeichnung enthält.

Mailadresse und **Firmenhomepage** werden mit Hilfe von Regulären Ausdrücken extrahiert.

> **Mail-Pattern:** *[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+.[a-zA-Z0-9-.]+*
>
> **WWW-Pattern:** *www.[\w\d.:#@%/;$()~_?+-=&]**
>
> **HTTP-Pattern:** *http.[\w\d.:#@%/;$()~_?+-=&]**

[3] **www.stepstone.de**.
[4] **www.monster.de**.

HTTPS-Pattern: *https.[\w\d.:#@%/;$()~_?+-=&]**

Leider können Reguläre Ausdrücke nur Daten mit einer eindeutigen Struktur ermitteln. **Kontaktperson** und **Firmenadresse** können aber sehr unterschiedlich geschrieben werden:

- Alle der Kosmonauten 26a D-32451 Bad Kreuznach
- Straße des 17. Juni 17a 13456 Berlin (Lieblings Beispiel)
- Herr Richard-Alexander Wagner
- Johann Wolfgang von Goethe
- Alexander Sergejewitsch Puschkin

Aus den oben aufgelisteten Beispielen wird es ersichtlich, dass es kaum möglich ist, allein mit Regulären Ausdrücke, diese Daten zu extrahieren. Deswegen werden diese Daten in *„IntelliJob"* mit Hilfe von *„Apache OpenNLP Framework"*[5] gefunden und zusammen mit allen anderen bewerbungsrelevanten Daten in Datenbank gespeichert.

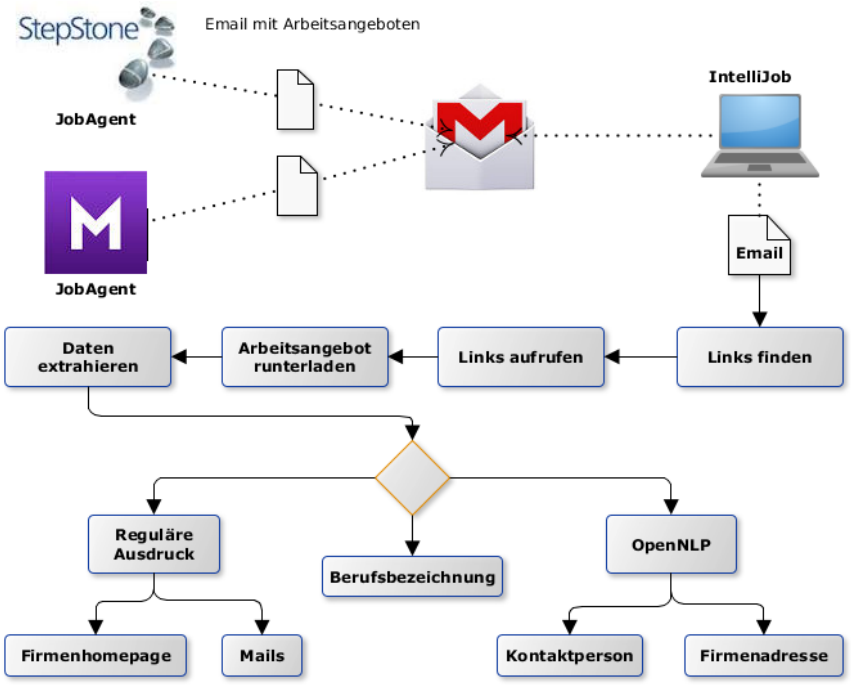

Abbildung 2.4: Prozessablauf in IntelliJob

[5] **https://opennlp.apache.org**.

2.2.1 Technologie Stack

Apache OpenNLP ist ein Open Source Produkt und wird als Maven-Dependency in der Version 1.5.3 in das Projekt *„IntelliJob"* eingebunden. Das Framework setzt die Technologie aus dem Gebiet *Natural Language Processing* ein, um bestimmte Bestandteile eines Textes zu erkennen und zu klassifizieren. Die Muster-Erkennung basiert auf künstlich erzeugten Trainingsdaten, die durch ein Lernprozess in einem Model gespeichert werden. Das Framework wird ständig weiter entwickelt und verfügt über eine gute Dokumentation.[6]

Die Anwendungslogik von *„IntelliJob"* wurde in der Programmiersprache Java geschrieben und verwendet sehr viele Komponenten aus dem Spring Framework[7]. Spring Framework ist ebenfalls ein Open Source Produkt, welches performante und moderne Java-Enterprise Einsätze realisiert. Mit dem Einsatz von Spring *Dependency Injection* und *Aspekt orientierte Programmierung* wird das Programmcode redundant verteilt und kann auch leicht wiederverwendet werden. Alle erzeugte Objekte werden anhand der *Spring Data*[8] in eine dokumentenorientierte Datenbank MongoDB persistiert.

Die weitere Besonderheit von *„IntelliJob"* ist, dass es eine Standalone Web-Anwendung ist, die lokal auf dem Client einen embedded Web Container *Tomcat*[9] startet. Der große Vorteil von diesem Einsatz liegt darin, dass der Webserver ein Teil der Anwendung ist, und die Anwendung nicht mehr in den Webserver deployed werden muss. Das Programm wird als JAR-Datei ausgeliefert und kann direkt in der Java Virtual Machine ausgeführt werden. Realisiert wird das ganze mit *Spring Boot*[10] Technologie, die dank Autokonfiguration sowohl alle nötige Resource zusammenpackt, als auch alle Rest-Services zur Kommunikation mit Frontend initialisiert.

Das Frontend wird mit *AngualrJS*[11] umgesetzt und verfügt bereits über 6 verschiedene Views. Das Framework unterstützt die Entwickler bei der Implementierung von Single Page Anwendungen.

Home	Startseite um Mail-Account auszuwählen und Zugangsdaten zu übergeben.
Emails	zeigt alle gefundene Emails von Job Portals
Job Links	zeigt alle in Mails gefundene Links zu Arbeitsangeboten
Jobs	zeigt alle runter geladene Arbeitsangebote
Jobs Details	zeigt alle extrahierte Daten eines Arbeitsangebotes
Audit	dient zur Datenauswertung und zeigt sowohl die aktuellste als auch die alte Ergebnisse des Datenextraktions.

Während der Masterarbeit wird *AngularJS* von der Version 1.2.16 zu 1.4.7 aktualisiert.

[6] **https://opennlp.apache.org/documentation/1.6.0/manual/opennlp.html**.
[7] **www.spring.io**.
[8] **http://projects.spring.io/spring-data**.
[9] **http://tomcat.apache.org**.
[10] **http://projects.spring.io/spring-boot**.
[11] **https://angularjs.org**.

2.2.2 Software Architektur

„IntelliJob" ist quelloffen und kann auf Versionsverwaltungsplattform, GitHub[12], heruntergeladen werden. Die Anwendungslogik basiert auf Schichtenarchitektur, die sequenziell von oben nach unten abgearbeitet wird.

Präsentationsschicht repräsentiert die Daten in HTML-Form und regelt die Interaktionen zwischen Benutzer und Software. Alle Benutzeranfragen werden asynchron an Serviceschicht weitergeleitet. Die Daten zwischen beiden Schichten werden in JSON-Format transportiert.

Serviceschicht nimmt die Client-Anfragen entgegen, validiert Benutzerdaten und wandelt sie in Domain-Objekte um. Danach wird die Abfrage der Controlschicht übergeben. Letztendlich wird ein *Response* generiert und zurückgesendet.

Controlschicht vereint alle Methoden aus Datenzugriffsschicht und führt alle nötige Operationen aus, um die Benutzeranfrage zu bearbeiten.

Datenzugriffsschicht kapselt die Zugriffe auf persistente Daten. Die Datenaustausch erfolgt über Domain-Objekte.

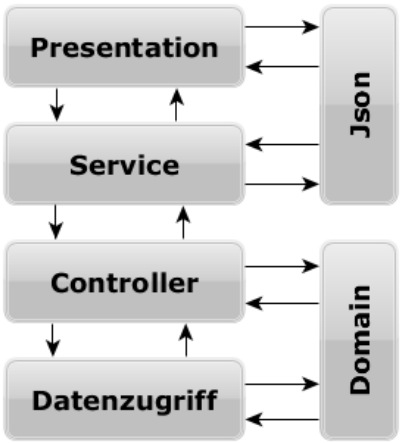

Abbildung 2.5: Schichtenarchitektur

Dadurch dass die Anwendungslogik in mehrere Schichten verteilt wird, wird die Komplexität der Anwendung wesentlich reduziert, was sowohl für das Verständnis als auch für die Wartung von Software eine große Vorteil ist. Außerdem können die Programmteile dank seiner Abstraktion gut getestet und auch leicht wiederverwendet werden.

Um die Methoden zum Datenextraktion auch in anderen Projekten zu verwenden, werden sie in ein separates Projekt, namens *Civis-Tools*[13] ausgelagert.

[12] **https://github.com/SergejMeister/intellijob**.
[13] **https://github.com/SergejMeister/civis-tools**.

2.2.3 Problembeschreibung

Zum Beginn der Masterarbeit gab es ein kleines Problem in *„IntelliJob*", das früher immer missachtet wurde und während der Masterarbeit unbedingt gelöst werden müsste. Das Problem liegt in doppelt gespeicherten Arbeitsangeboten.

Jedes Mail von Job-Portal beinhaltet Links sowohl mit neuen als auch mit alten Arbeitsangeboten. Deswegen wird es vor dem Schreiben in Datenbank immer geprüft, ob das Link zu diesem Arbeitsangebot schon früher gespeichert wurde. Durch die Beobachtung wurde allerdings erkannt, dass die Links zum gleichen Arbeitsangebot manchmal geändert werden. Außerdem kann ein Arbeitsangebot sowohl von StepStone als auch von Monster Portal kommen. Die Links und Mail sind dabei unterschiedlich und es gibt auch keine logische Möglichkeit die beiden Arbeitsangebote von einander zu unterscheiden.

2.3 Tag Cloud

Abbildung 2.6: Tag Clouds - *www.einfachbewusst.de*

Die Tag Cloud, deutscher Begriff Schlagwortwolke, werden im Web immer beliebter und dienen zur Visualisierung von gewichteten Daten. Je nach Gewichtung werden die Daten in verschiedenen Schriftgröße und Farben dargestellt. Mit Hilfe von Tag Cloud können beispielsweise die meistbesuchte Beitrage eines Weblogs oder die häufigste Worte eines Dokuments angezeigt werden. In der Implementierungsphase dieser Masterarbeit werden die Tag Clouds zur Visualisierung von persönlichen Fähigkeiten in der Suchmaske eingesetzt. Dadurch wird es für den Benutzer ersichtlich, welche Fähigkeiten bei der Suche besonders priorisiert werden.

In der Praxis werden die einzelnen Wörter der Tag Cloud mit einer Unterseite oder Dokument verlinkt, die am relevantesten für diese Schlagwörter sind. So entsteht eine Art alternative zu einem Navigationsmenü. Der Nachteil von diesem Einsatz besteht darin, dass die Gewichtung der Schlagwörter durch Aufruf der Seite oder Dokumentes erfolgt und keine Information darüber enthält, ob der verlinkte Inhalt den Benutzererwartungen entspricht. Der Benutzer klickt auf dem hoch gewichtete Schlagwort an, stellt fest, dass der verlinkte Inhalt nicht den Erwartungen

entspricht und verlässt die Seite. Da die Seite jedoch aufgerufen wurde, wird der Schlagwort der Tag Cloud höher gewichtet[OnPage 2016].

Die Tag Cloud wird meistens nach der folgenden Formel berechnet:

$$S_i = [(f_{max} - f_{min}) * \frac{t_i - t_{min}}{t_{max} - t_{min}} + fmin]$$

- S_i - anzuzeigende Schriftgröße
- f_{max} - maximale Schriftgröße
- f_{min} - minimale Schriftgröße
- t_i - Häufigkeit des betreffenden Schlagwortes
- t_{min} - Häufigkeit, ab der ein Schlagwort angezeigt werden soll
- t_{max} - Häufigkeit des häufigsten Schlagwortes[14]

2.4 Search Engine

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Lucene, Elasticsearch

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

[14] **https://de.wikipedia.org/wiki/Schlagwortwolke**.

Volltextsuche

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext" oder „Huardest gefburn"? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum" dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Gewichtete Suche

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext" oder „Huardest gefburn"? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum" dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Cosine similarity

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext" oder „Huardest gefburn"? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum" dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

3. Systementwurf

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

3.0.1 Prozessablauf

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

3.0.2 Domain-Schicht

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine

falsche Anmutung vermitteln.

3.0.3 DAO-Schicht

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext" oder „Huardest gefburn"? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum" dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

3.0.4 Controller-Schicht

Unbedingt beschreiben, dass das Controller auf civis-tools zugreift, um die Daten aus dem Text zu extrahieren.

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext" oder „Huardest gefburn"? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum" dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

3.0.5 Service-Schicht

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext" oder „Huardest gefburn"? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum" dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

3.0.6 Darstellung-Schicht

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

4. Entwicklung und Implementierung

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

4.1 Schnittstelle

?

4.2 Suchfunktionen

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

4.2.1 API

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest

gefburn"? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum" dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

4.2.2 Benutzeroberfläche

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext" oder „Huardest gefburn"? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum" dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

4.2.3 Softwarequalität

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext" oder „Huardest gefburn"? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum" dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

5. Evaluierung

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

6. Problemen und Schwierigkeiten

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

7. Zusammenfassung und Ausblick

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Literaturverzeichnis

Beringer, Steffen (2012). „Effizienz und Effektivität der Integration von Textklassifikation in Information Extraction-Pipelines". master thesis. Universität Paderborn (siehe S. 1).

Dorschel, Joachim (2015). „Praxishandbuch Big Data - Wirtschaft – Recht – Technik". 1. Aufl. Berlin Heidelberg New York: Springer-Verlag (siehe S. 4, 6).

Jan, Steemann (2013). *Datenmodellierung in nicht relationalen Datenbanken*. url: https://entwickler.de/online/datenbanken/datenmodellierung-in-nicht-relationalen-datenbanken-137872.html (siehe S. 6).

OnPage (2016). *Tag Cloud*. url: https://de.onpage.org/wiki/Tag_Cloud (siehe S. 12).

Abbildungsverzeichnis

Tabellenverzeichnis

Glossar

NoSQL - not only SQL - bezeichnet Datenbanken, die einen nicht-relationalen Ansatz verfolgen.

SentenceDetector - Machine Learning Technologie - teilt Text in einzelne Sätze

Tokenizer - Machine Learning Technologie - teilt Sätze in einzelne Worte

Named Entity Recognition - Machine Learning Technologie - erkennt und klassifiziert Bestandteile im Text.

Part-Of-Speech tagging - Machine Learning Technologie - Das Zuweisen von Markierungen zu einzelnen Einheiten (Wortart-Annotierung)

Chunker - Machine Learning Technologie - Teilt Text in syntaktisch korrelierten Teile von Wörtern, wie Nomen Gruppen.

Coreference Resolution - Machine Learning Technologie - Bezugnahme auf dieselbe Entität.

DAO - Data Access Object - Adapter zur Abstrahierung und Entkopplung von Datenzugriffe.

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel verfasst habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Datum: 24. Januar 2016, Berlin **Unterschrift:**